

IN THE CLAIMS:

Please amend the claims as follows:

- C1
1. (Currently Amended) A method of ~~processing~~ translating a source code statement written in a programming language into a markup language, the method comprising the computer-implemented steps of:
 - parsing a document type definition file for a markup language;
 - parsing said source code statement from a source code file;
 - selecting an element defined in the document type definition file based on an association between the element and an identifier of a routine in said source code statement;
 - and
 - writing the selected element to a markup language file to form a translation.
 2. (Original) The method of claim 1 wherein the source code statement comprises parameters for the routine and wherein the element comprises an attribute list corresponding to the parameters.
 3. (Original) The method of claim 2 wherein the selected element written to the markup language file comprises an attribute list of values for the parameters passed to the routine.
 4. (Original) The method of claim 1 wherein the routine is a procedure, subroutine, function, method, class, or software object.
 5. (Currently Amended) A method of ~~processing~~ translating a markup language element into a source code statement, the method comprising the computer-implemented steps of:
 - parsing a document type definition file for the markup language;
 - parsing a markup language element from a markup language file;
 - selecting an element defined in the document type definition file that is equivalent to the markup language element from the markup language file;
 - generating a source code statement using an identifier of a routine within the selected element; and
 - writing the source code statement to an output file to form a translation.

6. (Currently Amended) A method of generating dynamically translating an application program into a markup language file, the method comprising the computer-implemented steps of:
- executing an said application program;
 - parsing a document type definition file for a markup language;
 - during execution of said application program, selecting an element defined in the document type definition file based on a routine called by the said application program; and
 - writing the selected element to a markup language file to form a translation.
7. (Original) The method of claim 6 wherein the element comprises an attribute list corresponding to parameters for the routine.
8. (Original) The method of claim 6 wherein the selected element written to the markup language file comprises an attribute list corresponding to values for the parameters passed to the routine.
9. (Original) The method of claim 6 wherein the application program is written in Java programming language.
10. (Original) The method of claim 9 wherein the routine is an extended class method.
11. (Original) The method of claim 9 wherein the routine is a Graphics class method.
12. (Currently Amended) A data processing system for processing translating a source code statement written in a programming language into a markup language, the data processing system comprising:
- first parsing means for parsing a document type definition file for a markup language;
 - second parsing means for parsing a source code statement from a source code file;
 - selecting means for selecting an element defined in the document type definition file based on an association between the element and an identifier of a routine in the source code statement; and
 - writing means for writing the selected element to a markup language file to form a translation.

13. (Original) The data processing system of claim 12 wherein the source code statement comprises parameters for the routine and wherein the element comprises an attribute list corresponding to the parameters.

14. (Original) The data processing system of claim 13 wherein the selected element written to the markup language file comprises an attribute list of values for the parameters passed to the routine.

15. (Original) The data processing system of claim 12 wherein the routine is a procedure, subroutine, function, method, class, or software object.

16. (Currently Amended) A data processing system for processing translating a markup language element into a source code statement, the data processing system comprising:

first parsing means for parsing a document type definition file for the markup language;

second parsing means for parsing a markup language element from a markup language file;

selecting means for selecting an element defined in the document type definition file that is equivalent to the markup language element from the markup language file;

generating means for generating a source code statement using an identifier of a routine within the selected element; and

writing means for writing the source code statement to an output file to form a translation.

- 5/17/03
7. (Currently Amended) A data processing system for generating dynamically translating an application program into a markup language file, the data processing system comprising:
- executing means for executing an application program;
 - parsing means for parsing a document type definition file for a markup language;
 - selecting means for selecting an element defined in the document type definition file based on a routine called by the application program; and
 - writing means for writing the selected element to a markup language file to form a translation.
- 21
18. (Original) The data processing system of claim 17 wherein the element comprises an attribute list of parameters for the routine.
19. (Original) The data processing system of claim 17 wherein the selected element written to the markup language file comprises an attribute list of values for the parameters passed to the routine.
20. (Original) The data processing system of claim 17 wherein the application program is written in Java programming language.
21. (Original) The data processing system of claim 20 wherein the routine is an extended class method.
22. (Original) The data processing system of claim 20 wherein the routine is a Graphics class method.

23. (Currently Amended) A computer program product in a computer readable medium for use in a data processing system for ~~processing~~ translating a source code statement written in a programming language into a markup language, the computer program product comprising:

first instructions for parsing a document type definition file for a markup language;

second instructions for parsing a source code statement from a source code file;

third instructions for selecting an element defined in the document type definition file based on an association between the element and an identifier of a routine in the source code statement; and

fourth instructions for writing the selected element to a markup language file to form a translation.

24. (Currently Amended) A computer program product on a computer readable medium for use in a data processing system for ~~processing~~ translating a markup language element into a source code statement, the computer program product comprising:

first instructions for parsing a document type definition file for the markup language;

second instructions for parsing a markup language element from a markup language file;

third instructions for selecting an element defined in the document type definition file that is equivalent to the markup language element from the markup language file;

fourth instructions for generating a source code statement using an identifier of a routine within the selected element; and

fifth instructions for writing the source code statement to an output file to form a translation.

25. (Currently Amended) A computer program product on a computer readable medium for use in a data processing system for processing dynamically translating an application program into a markup language file, the computer program product comprising:

- first instructions for executing an application program;
- second instructions for parsing a document type definition file for a markup language;
- third instructions for selecting an element defined in the document type definition file based on a routine called by the application program; and
- fourth instructions for writing the selected element to a markup language file to form a translation.

26. (Currently Amended) A method of processing translating a source code statement written in a programming language into a markup language, the method comprising the computer-implemented steps of:

- parsing a grammar input file for a markup language;
- parsing a source code statement from a source code file;
- selecting a language syntax construct defined in the grammar input file based on an association between the language syntax construct and an identifier of a routine in the source code statement; and
- writing the selected language syntax construct to a markup language file to form a translation.